WHAT IS CLAIMED:

1. A mathematics coprocessor comprising:

a multiplier - accumulator unit comprising:

a multiplier array for selectively multiplying first and second operands, the first and second operands having a data type selected from the group including floating point and integer data types;

an adder for selectively performing addition and subtraction operations on third and fourth operands; and

multiplexer circuitry for selectively presenting the third and fourth operands to inputs of the adder, the multiplexer circuitry selecting the third and fourth operands from the contents of a set of associated source registers, data output from the multiplier array and data output from the adder.

2. The coprocessor of Claim 1 wherein the third and fourth operands comprise integers.

3. The coprocessor of Claim 1 wherein said multiply - accumulate unit is operable during a double precision multiplication to:

multiply an unsigned first set of bits from a first source register with an unsigned first set of bits from a second source register to generate a first product and first carry bit;

add the first product and first carry bit with a first constant to generate a first sum;

multiply the unsigned first set of bits from the first source register with an unsigned second set of bits from the second source register to generate a second product and second carry bit;

add the second product and carry bit with the first sum to generate a second sum;

multiply a signed second set of bits from the first source register with the unsigned first set of bits from the second register to generate a third product and carry bit;

add the second sum with the third product and carry bit to generate a third sum;

multiply the signed second set of bits from the first source register with the signed second set of bits from the second source register to generate a fourth product; and

add the fourth product with the third sum, the third sum being selectively shifted, to generate a product of the contents of the first and second source registers.

B1 Cont.

4. The coprocessor of Claim 3 wherein the first set of bits from the first and second source registers each comprise an upper set of bits from integers stored in the first and second registers.

5. The coprocessor of Claim 3 wherein the first set of bits from the first and second source registers comprise an upper set of bits from mantissas stored in the first and second registers.

6. The coprocessor of Claim 3 wherein the second set of bits from the first and second source registers comprise a lower set of bits from integers stored in the first and second source registers.

7. The coprocessor of Claim 3 wherein the second set of bits from the first and second source registers comprise a lower set of bits from mantissas stored in the first and second source registers.

8. The coprocessor of Claim 1 and further comprising a floating point comparator for selectively comparing operands presented in a set of source registers.

B1 Cont.

9. The coprocessor of Claim 1 and further comprising a floating point adder for performing floating point addition and subtraction operations on operands presented in a set of source registers.

B1 cont'd

10. A digital signal processor comprising:

a multiplier-accumulator for performing integer and floating point multiplication and integer addition operations on operands selectively fetched into a set of source registers;

a floating point adder for performing floating point addition operations on operands selectively fetched into the set of source registers; and

a comparator for comparing floating point operands selectively fetched into the set of source registers.

11. The digital signal processor of Claim 10 wherein said multiplier - accumulator unit comprises:

a multiplier array for selectively multiplying floating point mantissas and integers;

an fixed point adder for selectively performing addition operations on data including integers received from the set of source registers and products generated by the multiplier array; and

an accumulator including a register for accumulating results generated by the fixed point adder.

12. The digital signal processor of Claim 11 wherein said multiplier - accumulator further comprises a shift register for selectively shifting data including operands received from the set of source registers and results generated by the fixed point adder.

B1 cont'd

13. The digital signal processor of Claim 11 wherein said digital signal processor comprises a math coprocessor operating in conjunction with a microprocessor.

14. The digital signal processor of Claim 11 wherein said digital signal processor comprises a coprocessor operating in conjunction with a reduced instruction set computer.

15. The digital signal processor of Claim 11 wherein said multiplier - accumulator further comprises circuitry for selectively forwarding results directly to said floating point adder to prevent pipeline bubbles.

16. The digital signal processor of Claim 11 wherein said floating point adder comprises circuitry for selectively forwarding results directly to said multiplier - accumulator to prevent pipeline bubbles.

17. The digital signal processor of Claim 11 wherein said multiplier - accumulator comprises:

a multiplier array for multiplying first and second operands during a first clock period;

a fix point adder for adding a result from said multiplier array with a third operand during a second clock period; and

an accumulator register for storing a sum output from said adder during the second clock period.

B1 cont'd.

18. A method of performing arithmetic operations in a multiplier operable to perform both integer and floating point operations comprising the steps of:

in response to a first instruction, performing a single precision multiplication of first and second signed floating point operands comprising the substeps of:

adding exponents of the first and second operands;

multiplying a signed mantissa of each of the operands in a multiplier array to generate a product and a carry bit;

adding the partial product and carry bit with a constant using a fixed point adder to generate an intermediate result;

selectively rounding and renormalizing the intermediate result; and

in response to a second instruction, performing a single precision multiplication of first and second integers comprising the substeps of:

multiplying the signed first and second integers in the multiplier array to generate a product and a carry bit;

adding the product and carry bit with a constant using the fixed point adder to generate an intermediate result; and

selectively rounding and renormalizing the intermediate result.

19. The method of Claim 18 and further comprising the step of performing a double precision multiplication of first and second floating point operands in response to a third instruction comprising the substeps of:

adding exponents of the first and second operands;

multiplying unsigned lower bits of a mantissa of the first operand with unsigned lower bits of a mantissa of the second operand in the multiplier array to generate a first partial product and a carry bit;

adding the first partial product and carry bit with a constant using the fixed point adder to generate first intermediate result;

multiplying the unsigned lower bits of the mantissa of the first operand with unsigned upper bits of the mantissa of the second operand in the multiplier array to generate a second partial product and second carry bit;

selectively shifting the first intermediate result by a selected shift count;

adding the second partial product and second carry bit with the shifted first intermediate result using the fixed point adder to generate a second intermediate result;

multiplying signed upper bits of the mantissa of the first operand with the unsigned lower bits of the mantissa of the second operand in the multiplier array to generate a third partial product and third carry bit;

adding the third partial product and third carry bit with the second intermediate result using the fixed point adder to generate a third intermediate result;

multiplying the signed upper bits of the mantissa of

31 cont'd.

the first operand with the signed upper bits of the mantissa of the second operand in the multiplier array to generate a fourth partial product and fourth carry bit;

selectively shifting the third intermediate result by a selected shift count;

adding the fourth partial product and forth carry bit with the shifted third intermediate result using the fixed point adder to generate a fourth intermediate result; and

selectively rounding and renormalizing the fourth intermediate result to generate a final product.

B1 cont.

20. The method of Claim 18 and further comprising the step of performing a double precision multiplication on first and second signed integers comprising the substeps of:

multiplying unsigned lower bits of the first and second integers in the multiplier array to generate a first partial product and first carry bit;

adding the first partial product and first carry bit with a constant using the fixed point adder to generate a first intermediate result;

multiplying the unsigned lower bits of the first integer and unsigned upper bits of the second integer in the multiplier array to generate a second product and second carry bit;

selectively shifting the first intermediate result by a selected shift count;

adding the second partial product and second carry bit with the shifted first intermediate result using the fixed point adder to generate a second intermediate result;

multiplying signed upper bits of the first integer with the unsigned lower bits of the second integer in the multiplier array to generate a third partial product and third carry bit;

adding the second intermediate result with the third partial produce and third carry bit using the fixed point adder to generate a third intermediate result;

multiplying the signed upper bits of the first and second integers in the multiplier array to generate a fourth partial product and fourth carry bit;

Bl cont.

shifting the third intermediate result by a selected shift count;

adding the shifted third intermediate result with the fourth partial product and fourth carry bit in the fixed point adder to generate a fourth intermediate result; and

selectively rounding and renormalizing the fourth intermediate result to generate a final product.

21. The method of Claim 18 and further comprising the step of adding first and second integers in the multiplier in response to a third instruction comprising the steps of:

presenting the first and second integers to corresponding inputs of the fixed point adder forming a portion of the multiplier; and

adding the first and second integers with the fixed point adder.

22. The method of Claim 18 wherein the multiplier further includes at least one accumulator and said step of performing a single precision integer multiplication further comprises the substeps of:

adding a third integer to the intermediate result to generate a sum;

storing the sum in the accumulator.

23. The method of Claim 18 wherein the multiplier further includes at least one accumulator and said step of performing a single precision integer multiplication further comprises the substeps of:

subtracting a third integer from the intermediate result to generate a result; and

storing the result in the accumulator.

24. The method of Claim 18 wherein the multiplier further includes at least one accumulator and said step of performing a single precision integer multiplication further comprises the substeps of:

adding the intermediate result to a value stored in an accumulator; and

storing the result of said substep of adding in an accumulator.

25. The method of Claim 18 wherein the multiplier further includes at least one accumulator and said step of performing a single precision integer multiplication further comprises the substeps of:

subtracting a value stored in an accumulator from the intermediate result; and

storing the result of said substep of subtracting in an accumulator.

31 cont.

26. An instruction set for operating a processor including a multiplier array, a fixed point adder and a floating point adder comprising:

a first set of instructions for multiplying first and second operands, at least some bits of each of said first and second operands multiplied in said multiplier array and a result of the multiplication added to a third value by the fixed point adder;

a second set of instructions for adding first and second integers using said fixed point adder; and

a third set of instructions for adding first and second floating point values in said floating point adder.

27. The instruction set of Claim 26 wherein said first set of instructions comprises:

at least one instruction for multiplying first and second integer operands using said multiplier array and said fixed point adder; and

at least one instruction for multiplying first and second floating point operands using said multiplier array and said fixed point adder.

28. The instruction set of Claim 26 wherein said first set of instructions comprise:

at least one instruction for performing a double precision multiplication of said first and second operands; and

at least one instruction for performing a single precision multiplication of said first and second operands.

29. The instruction set of Claim 26 and further comprising a set of instructions for converting data between first and second data types.

30. The instruction set of Claim 26 wherein said first data type comprises floating point data and said second data type comprises integer data.

31. The instruction set of Claim 26 wherein said first data type comprises single precision data and said second data type comprises double precision data.

32. The instruction set of Claim 26 and further comprising a set of instructions for shifting data in a selected direction by a selected number of bits.

B1 Contd.

33. The instruction set of Claim 26 and further comprising a set of instructions for selectively comparing first and second floating point numbers in a floating point comparator circuit.

34. The instruction set of Claim 26 and further comprising a set of instructions for taking an absolute value of a selected operand.

35. The instruction set of Claim 26 and further comprising a set of instructions for negating the value of a selected operand.